The Next Generation Web Servers

Mou-Liang Kung
Department of Computer Science
Norfolk State University
700 Park Avenue
Norfolk, VA 23504
757-823-8650
757-823-9229 (fax)
mlkung@nsu.edu

Presentation Narrative

Web servers traditionally were used for publishing documents on the Internet. Most web documents are coded in HTML (the HyperText Markup Language). HTML defines "tags" which mark the text format and rendering instruction instead of document structures (e.g. authors, version, abstract, chapters, sections, ... etc). Web browser vendors obligingly implement the rendition of the HTML tags by hard-coding them into their HTML browsers, making them very hard to add new tags without sacrificing interoperability. Vendor-added tags are typical problems for interoperability. There is simple no extensibility to HTML at all. Therefore, tricks and work-around have been conceived to enhance web pages by adding Java applets and client-side scripts (e.g. VBScript, JavaScript, ...etc.) making these web pages dependent upon specific programming languages.

At the same time, more and more businesses are opening accesses to their corporate databases via web servers to conduct e-commerce (e.g., Internet advertising, Internet purchases). When it comes to business to business data exchange, a major problem arises. Since HTML is only a presentation language, not a structural one, businesses who wish to conduct business data exchange must open two browser windows and manually transcribe or copy-and-paste data from one web page to another to achieve data exchange. Manual operations are time-consuming and prone ë)ırrors. gàey are totally unsuitable for financial transactions or database information exchange. Therefore, HTML is inadequate to foster process automation. Hence, new generations of web servers are needed to escape the shortcomings of HTML while remain backward compatible to legacy HTML documents.

Since the World Wide Web Consortium published the XML ([1], eXtensible Markup Language) specification in December of 1997, XML has been touted as the technology to anchor the next-generation Web. XML is a lean subset of SGML ([2], Standard Generalized Markup Language, ISO 8879) for network document delivery. While SGML is a system and software independent meta-language used to specify document types and structures, so is XML. Document components tagged with XML-based structural elements can be easily and consistently parsed for electronic processing. The instant fame of XML can be attributed to its powerful capabilities not found in HTML:

- allowing different content views of a document to different users on different rendering devices,
- providing a language-independent, platform-neutral, structured data exchange format for web applications or heterogeneous databases for process automation and integration,
- Configuring and representing distributed components (written in Java or other OO languages) in CORBA ([3], Common Object Request Broker Architecture) distributed environment.

On the Internet, two businesses can exchange data in XML format directly between two XML-enabled web servers which serve as front-ends to their individual corporate database servers without human intervention. Hence, XML is good for consumer to business as well as business to business applications. Therefore, the next generation of web servers must be able to handle XML.

Prior to the birth of XML, these capabilities might be found in more expensive and complicated SGML-based electronic publishing systems. However, due to the simplicity of XML, there is a plethora of low cost or free XML tools ready for web sites to be enabled with XML handling capability. XML documents

can be created either by using plain text editors (e.g. Microsoft NotePad) or XML editors (e.g. XMetal, [4]) which simplify chores of following their document type definitions (DTD). The web servers must then be able to serve XML documents to XML browsers of the future as well as HTML browsers at present. There are at least two ways to deal with the XML to HTML conversion issue:

1. Use scripting language to offer CGI script hyperlinks. For example, one can use a Perl script to do the conversion.

2. Use Java servlet engine on web servers to implement Java servlets that will convert XML to HTML on the fly.

Currently, automated conversion can be done using the Cocoon servlet ([5]) on Apache JServ servlet engine ([6]), with an XML parser (e.g., IBM XML4J, [7]) and an XSL ([8], XML Stylesheet Language) processor (e.g., XSL:P Processor, [9]). XML parsers with DOM ([10], Document Object Model) support are essential since they parse the XML documents into document objects and validate them for further processing. The XSL processors with DOM support then attach styling information to these document objects to generate HTML output.

The experience of enabling web servers to serve XML documents to both traditional HTML browsers and browsers with some XML capabilities (e.g. Microsoft IE5, [11]) at Norfolk State University will be presented. A demo page can be found at http://199.111.112.120/~randy.

References

[1] Extensible Markup Language, http://www.w3.org/XML/
[2] Standard Generalized Markup Language, http://www.oasis-open.org/cover
[3] Common Object Request Broker Architecture
[4] XMetal, http://www.softquad.com/products/xmetal/
[5] Cocoon, http://java.apache.org/cocoon/
[6] Apache JServ servlet engine, http://java.apache.org/
[7] IBM XML4J, http://www.alphaworks.ibm.com/tech/xml4j
[8] Extensible Stylesheet Language, http://www.w3.org/Style/XSL/
[9] XSL:P Processor, http://www.clc-marketing.com/xslp/
[10] Document Object Model, http://www.w3.org/DOM/
[11] Microsoft IE5, http://www.microsoft.com/msdownload/iebuild/ie5_win32/